**Clases y objetos**

**N° de práctica: 1**

|  |  |  |
| --- | --- | --- |
| **Nombre completo del alumno** | | **Firma** |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
| **N° de brigada:** | **Fecha de elaboración:** | **Grupo:** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Elaborado por:** | **Revisado por:** | **Autorizado por:** | **Vigente desde:** |
| **M.F. Gabriel Hurtado Chong**  **Ing. Cynthia Nelly Peña Belmont** | **M.F. Gabriel Hurtado Chong** | **Colegio de Programación** | **18-Feb-16** |

1. **Seguridad en la ejecución**

|  |  |  |
| --- | --- | --- |
|  | **Peligro o Fuente de energía** | **Riesgo asociado** |
| 1 | Tensión alterna | Electrocución |

1. **Objetivos de aprendizaje**
2. **Objetivos generales:** Entender la diferencia entre clase y objeto.
3. **Objetivos específicos:**

* Aprender a diseñar clases.
* Entender el funcionamiento de los constructores.
* Identificar los atributos y los métodos.
* Aprender a crear objetos.

1. **Introducción**

Una clase es una construcción que permite crear tipos personalizados de datos mediante la agrupación de diversas variables, métodos y eventos.

Una clase es como un plano. Define los datos y el comportamiento de un tipo de datos. Si la clase no se declara como estática, el código de cliente puede utilizarla mediante la creación de objetos o instancias que se asignan a una variable.

A través de una clase se define un tipo de objeto, pero no es propiamente un objeto. Un objeto es una entidad concreta basada en una clase y, a veces, se denomina instancia de una clase.

Las clases contienen atributos que son un tipo de datos que guarda información y también son considerados como metadatos.

Las clases también incluyen métodos, que son bloques de código que contienen una serie de instrucciones.

Cada vez que se crea un objeto, se llama al constructor de la clase. Una clase puede tener varios constructores que toman argumentos diferentes. Los constructores permiten al programador establecer valores predeterminados, limitar la creación de instancias y escribir código flexible y fácil de leer.

1. **Equipo y Material**



Computadora

1. **Desarrollo**
2. **Actividad 1**

**Diseño de una clase**

A partir de un objeto real, elija al menos 3 características que lo identifiquen y 2 acciones que definan su comportamiento.

Posteriormente, realice una abstracción para definir una clase o plantilla para ese tipo de objetos, que incluya un mínimo de 3 atributos de diferentes tipos (enteros, flotantes, cadenas, booleanos, etc.) y al menos 6 métodos que sean necesarios para escribir/leer los valores de los atributos y/o para interactuar con los objetos. Incluya también al menos 2 constructores para la clase.

1. **Actividad 2**

**Creación de objetos**

Mediante la creación de objetos, realice pruebas de escritorio para verificar el funcionamiento correcto de su clase y, de ser necesario, realice los ajustes y correcciones adecuados.

Las pruebas deben incluir, sin limitarse a ello, los siguientes puntos:

* Verificación de todos los constructores implementados.
* Comprobación de la correcta asignación de valores a cada uno de los atributos definidos.
* Verificación de todos los métodos programados.

1. **Bibliografía**

* CEBALLOS SIERRA, Francisco Javier. **Microsoft C#. Curso de programación**. México, Alfaomega, 2007
* DEITEL, Harvey y Deitel, PAUL. **C# Cómo programar**.  España, Pearson, 2007
* LÓPEZ ROMÁN, Leobardo. **Metodología de la programación orientada a objetos**. México, Alfaomega, 2007